

Unidade VI:


Ordenação Interna - Counting Sort



PUC Minas

Adaptação dos slides elaborados pelo Instituto de Ciências Exatas e
Informática - Departamento de Ciência da Computação

- Funcionamento básico
- Algoritmo
- Análise do número de movimentações e comparações

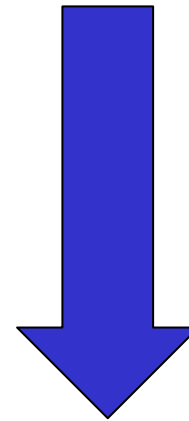
- **Funcionamento básico** 
- Algoritmo
- Análise do número de movimentações e comparações

Ideia Básica

- Triplicamos o número de *arrays* (entrada, contagem e saída)

Array de entrada
(a ser ordenado)

0	1	2	3	4	5	6	7



Array de contagem
(mapeamento de elementos)

Array de saída
(ordenado)

0	1	2	3	4	5	6	7

Ideia Básica

- Cada posição do array contagem armazena o número de elementos menores ou iguais a ela no array entrada. Por exemplo, se a entrada tem 3 zeros, 1 um e 2 dois, então o contagem tem 3, 4 e 6, respectivamente

Array de entrada

0		a	b	c		d	e
1	...	2	0	2	...	0	0

Array de contagem

0	1	2	3	4	5
3	4	6			

Array de entrada

0	1	2	3	4	5	6	7
2	5	3	0	2	3	0	3

Array de entrada

0	1	2	3	4	5	6	7
2	5	3	0	2	3	0	3

O *array* de contagem terá **seis** posições (0 à 5), porque o maior valor é 5.

O *array* de saída terá oito posições

Array de entrada

0	1	2	3	4	5	6	7
2	5	3	0	2	3	0	3

Array de contagem

0	1	2	3	4	5

Array de saída

0	1	2	3	4	5	6	7

Array de entrada

0	1	2	3	4	5	6	7
2	5	3	0	2	3	0	3

Array de contagem

0	1	2	3	4	5

Array de entrada

0	1	2	3	4	5	6	7
2	5	3	0	2	3	0	3

Array de contagem

0	1	2	3	4	5
0	0	0	0	0	0

Inicializar todas as posições
do *array* de contagem com zero

Array de entrada

0	1	2	3	4	5	6	7
2	5	3	0	2	3	0	3

Array de contagem

0	1	2	3	4	5
0	0	0	0	0	0

Para cada elemento do *array* de entrada,
incrementá-lo no de contagem

Array de entrada

0	1	2	3	4	5	6	7
2	5	3	0	2	3	0	3

Array de contagem

0	1	2	3	4	5
0	0	1	0	0	0

Para cada elemento do *array* de entrada,
incrementá-lo no de contagem

Array de entrada

0	1	2	3	4	5	6	7
2	5	3	0	2	3	0	3

Array de contagem

0	1	2	3	4	5
0	0	1	0	0	1

Para cada elemento do *array* de entrada,
incrementá-lo no de contagem

Array de entrada

0	1	2	3	4	5	6	7
2	5	3	0	2	3	0	3

Array de contagem

0	1	2	3	4	5
0	0	1	1	0	1

Para cada elemento do *array* de entrada,
incrementá-lo no de contagem

Array de entrada

0	1	2	3	4	5	6	7
2	5	3	0	2	3	0	3

Array de contagem

0	1	2	3	4	5
1	0	1	1	0	1

Para cada elemento do *array* de entrada,
incrementá-lo no de contagem

Array de entrada

0	1	2	3	4	5	6	7
2	5	3	0	2	3	0	3

Array de contagem

0	1	2	3	4	5
1	0	2	1	0	1

Para cada elemento do *array* de entrada,
incrementá-lo no de contagem

Array de entrada

0	1	2	3	4	5	6	7
2	5	3	0	2	3	0	3

Array de contagem

0	1	2	3	4	5
1	0	2	2	0	1

Para cada elemento do *array* de entrada,
incrementá-lo no de contagem

Array de entrada

0	1	2	3	4	5	6	7
2	5	3	0	2	3	0	3

Array de contagem

0	1	2	3	4	5
2	0	2	2	0	1

Para cada elemento do *array* de entrada,
incrementá-lo no de contagem

Array de entrada

0	1	2	3	4	5	6	7
2	5	3	0	2	3	0	3

Array de contagem

0	1	2	3	4	5
2	0	2	3	0	1

Para cada elemento do *array* de entrada,
incrementá-lo no de contagem

Array de entrada

0	1	2	3	4	5	6	7
2	5	3	0	2	3	0	3

Array de contagem

0	1	2	3	4	5
2	0	2	3	0	1

Array de entrada

0	1	2	3	4	5	6	7
2	5	3	0	2	3	0	3

Array de contagem

0	1	2	3	4	5
2	0	2	3	0	1

Fazer com que o *array* de contagem seja acumulativo de tal forma que cada posição i armazene o número de elementos menores ou iguais a i

Array de entrada

0	1	2	3	4	5	6	7
2	5	3	0	2	3	0	3

Array de contagem

0	1	2	3	4	5
2	0	2	3	0	1

Fazer com que o *array* de contagem seja acumulativo de tal forma que cada posição i armazene o número de elementos menores ou iguais a i

Array de entrada

0	1	2	3	4	5	6	7
2	5	3	0	2	3	0	3

Array de contagem

0	1	2	3	4	5
2	2	2	3	0	1

Fazer com que o *array* de contagem seja acumulativo de tal forma que cada posição i armazene o número de elementos menores ou iguais a i

Array de entrada

0	1	2	3	4	5	6	7
2	5	3	0	2	3	0	3

Array de contagem

0	1	2	3	4	5
2	2	4	3	0	1

Fazer com que o *array* de contagem seja acumulativo de tal forma que cada posição i armazene o número de elementos menores ou iguais a i

Array de entrada

0	1	2	3	4	5	6	7
2	5	3	0	2	3	0	3

Array de contagem

0	1	2	3	4	5
2	2	4	7	0	1

Fazer com que o *array* de contagem seja acumulativo de tal forma que cada posição i armazene o número de elementos menores ou iguais a i

Array de entrada

0	1	2	3	4	5	6	7
2	5	3	0	2	3	0	3

Array de contagem

0	1	2	3	4	5
2	2	4	7	7	1

Fazer com que o *array* de contagem seja acumulativo de tal forma que cada posição i armazene o número de elementos menores ou iguais a i

Array de entrada

0	1	2	3	4	5	6	7
2	5	3	0	2	3	0	3

Array de contagem

0	1	2	3	4	5
2	2	4	7	7	8

Fazer com que o *array* de contagem seja acumulativo de tal forma que cada posição i armazene o número de elementos menores ou iguais a i

Array de entrada

0	1	2	3	4	5	6	7
2	5	3	0	2	3	0	3

Array de contagem

0	1	2	3	4	5
2	2	4	7	7	8

Preencher o *array* de saída, copiando os elementos da entrada de trás para frente nas suas respectivas posições

Array de entrada

0	1	2	3	4	5	6	7
2	5	3	0	2	3	0	3

Array de contagem

0	1	2	3	4	5
2	2	4	7	7	8

Array de saída

0	1	2	3	4	5	6	7

Preencher o *array* de saída, copiando os elementos da entrada de trás para frente nas suas respectivas posições

Array de entrada

0	1	2	3	4	5	6	7
2	5	3	0	2	3	0	3

Array de contagem

0	1	2	3	4	5
2	2	4	7	7	8

Array de saída

0	1	2	3	4	5	6	7

Preencher o *array* de saída, copiando os elementos da entrada de trás para frente nas suas respectivas posições

Array de entrada

0	1	2	3	4	5	6	7
2	5	3	0	2	3	0	3

Array de contagem

0	1	2	3	4	5
2	2	4	7	7	8

7ª posição

Array de saída

0	1	2	3	4	5	6	7

Preencher o *array* de saída, copiando os elementos da entrada de trás para frente nas suas respectivas posições

Array de entrada

0	1	2	3	4	5	6	7
2	5	3	0	2	3	0	3

Array de contagem

0	1	2	3	4	5
2	2	4	7	7	8

7ª posição

Array de saída

0	1	2	3	4	5	6	7
						3	

Preencher o *array* de saída, copiando os elementos da entrada de trás para frente nas suas respectivas posições

Array de entrada

0	1	2	3	4	5	6	7
2	5	3	0	2	3	0	3

Array de contagem

0	1	2	3	4	5
2	2	4	6	7	8

Atualizar *array* de contagem

Array de saída

0	1	2	3	4	5	6	7
						3	

Preencher o *array* de saída, copiando os elementos da entrada de trás para frente nas suas respectivas posições

Array de entrada

0	1	2	3	4	5	6	7
2	5	3	0	2	3	0	3

Array de contagem

0	1	2	3	4	5
2	2	4	6	7	8

Array de saída

0	1	2	3	4	5	6	7
						3	

Preencher o *array* de saída, copiando os elementos da entrada de trás para frente nas suas respectivas posições

Array de entrada

0	1	2	3	4	5	6	7
2	5	3	0	2	3	0	3

Array de contagem

0	1	2	3	4	5
2	2	4	6	7	8

2ª posição

Array de saída

0	1	2	3	4	5	6	7
						3	

Preencher o *array* de saída, copiando os elementos da entrada de trás para frente nas suas respectivas posições

Array de entrada

0	1	2	3	4	5	6	7
2	5	3	0	2	3	0	3

Array de contagem

0	1	2	3	4	5
2	2	4	6	7	8

2ª posição

Array de saída

0	1	2	3	4	5	6	7
	0					3	

Preencher o *array* de saída, copiando os elementos da entrada de trás para frente nas suas respectivas posições

Array de entrada

0	1	2	3	4	5	6	7
2	5	3	0	2	3	0	3

Array de contagem

0	1	2	3	4	5
1	2	4	6	7	8

Atualizar *array* de contagem

Array de saída

0	1	2	3	4	5	6	7
	0					3	

Preencher o *array* de saída, copiando os elementos da entrada de trás para frente nas suas respectivas posições

Array de entrada

0	1	2	3	4	5	6	7
2	5	3	0	2	3	0	3

Array de contagem

0	1	2	3	4	5
1	2	4	6	7	8

Array de saída

0	1	2	3	4	5	6	7
	0					3	

Preencher o *array* de saída, copiando os elementos da entrada de trás para frente nas suas respectivas posições

Array de entrada

0	1	2	3	4	5	6	7
2	5	3	0	2	3	0	3

Array de contagem

0	1	2	3	4	5
1	2	4	6	7	8

6ª posição

Array de saída

0	1	2	3	4	5	6	7
	0					3	

Preencher o *array* de saída, copiando os elementos da entrada de trás para frente nas suas respectivas posições

Array de entrada

0	1	2	3	4	5	6	7
2	5	3	0	2	3	0	3

Array de contagem

0	1	2	3	4	5
1	2	4	6	7	8

6ª posição

Array de saída

0	1	2	3	4	5	6	7
	0				3	3	

Preencher o *array* de saída, copiando os elementos da entrada de trás para frente nas suas respectivas posições

Array de entrada

0	1	2	3	4	5	6	7
2	5	3	0	2	3	0	3

Array de contagem

0	1	2	3	4	5
1	2	4	5	7	8

Atualizar array de contagem

Array de saída

0	1	2	3	4	5	6	7
	0				3	3	

Preencher o *array* de saída, copiando os elementos da entrada de trás para frente nas suas respectivas posições

Array de entrada

0	1	2	3	4	5	6	7
2	5	3	0	2	3	0	3

Array de contagem

0	1	2	3	4	5
1	2	4	5	7	8

Array de saída

0	1	2	3	4	5	6	7
	0				3	3	

Preencher o *array* de saída, copiando os elementos da entrada de trás para frente nas suas respectivas posições

Array de entrada

0	1	2	3	4	5	6	7
2	5	3	0	2	3	0	3

Array de contagem

0	1	2	3	4	5
1	2	4	5	7	8

Array de saída

0	1	2	3	4	5	6	7
	0				3	3	

Preencher o *array* de saída, copiando os elementos da entrada de trás para frente nas suas respectivas posições

Array de entrada

0	1	2	3	4	5	6	7
2	5	3	0	2	3	0	3

Array de contagem

0	1	2	3	4	5
1	2	4	5	7	8

4ª posição

Array de saída

0	1	2	3	4	5	6	7
	0		2		3	3	

Preencher o *array* de saída, copiando os elementos da entrada de trás para frente nas suas respectivas posições

Array de entrada

0	1	2	3	4	5	6	7
2	5	3	0	2	3	0	3

Array de contagem

0	1	2	3	4	5
1	2	3	5	7	8

Atualizar *array* de contagem

Array de saída

0	1	2	3	4	5	6	7
	0		2		3	3	

Preencher o *array* de saída, copiando os elementos da entrada de trás para frente nas suas respectivas posições

Array de entrada

0	1	2	3	4	5	6	7
2	5	3	0	2	3	0	3

Array de contagem

0	1	2	3	4	5
1	2	3	5	7	8

Array de saída

0	1	2	3	4	5	6	7
	0		2		3	3	

Preencher o *array* de saída, copiando os elementos da entrada de trás para frente nas suas respectivas posições

Array de entrada

0	1	2	3	4	5	6	7
2	5	3	0	2	3	0	3

Array de contagem

0	1	2	3	4	5
1	2	3	5	7	8

1ª posição



Array de saída

0	1	2	3	4	5	6	7
	0		2		3	3	

Preencher o *array* de saída, copiando os elementos da entrada de trás para frente nas suas respectivas posições

Array de entrada

0	1	2	3	4	5	6	7
2	5	3	0	2	3	0	3

Array de contagem

0	1	2	3	4	5
1	2	3	5	7	8

1ª posição

Array de saída

0	1	2	3	4	5	6	7
0	0		2		3	3	

Preencher o *array* de saída, copiando os elementos da entrada de trás para frente nas suas respectivas posições

Array de entrada

0	1	2	3	4	5	6	7
2	5	3	0	2	3	0	3

Array de contagem

0	1	2	3	4	5
0	2	3	5	7	8

Atualizar array de contagem

Array de saída

0	1	2	3	4	5	6	7
0	0		2		3	3	

Preencher o *array* de saída, copiando os elementos da entrada de trás para frente nas suas respectivas posições

Array de entrada

0	1	2	3	4	5	6	7
2	5	3	0	2	3	0	3

Array de contagem

0	1	2	3	4	5
0	2	3	5	7	8

Array de saída

0	1	2	3	4	5	6	7
0	0		2		3	3	

Preencher o *array* de saída, copiando os elementos da entrada de trás para frente nas suas respectivas posições

Array de entrada

0	1	2	3	4	5	6	7
2	5	3	0	2	3	0	3

Array de contagem

0	1	2	3	4	5
0	2	3	5	7	8

Array de saída

0	1	2	3	4	5	6	7
0	0		2		3	3	

Preencher o *array* de saída, copiando os elementos da entrada de trás para frente nas suas respectivas posições

Array de entrada

0	1	2	3	4	5	6	7
2	5	3	0	2	3	0	3

Array de contagem

0	1	2	3	4	5
0	2	3	5	7	8

5ª posição

Array de saída

0	1	2	3	4	5	6	7
0	0		2		3	3	

Preencher o *array* de saída, copiando os elementos da entrada de trás para frente nas suas respectivas posições

Array de entrada

0	1	2	3	4	5	6	7
2	5	3	0	2	3	0	3

Array de contagem

0	1	2	3	4	5
0	2	3	5	7	8

5ª posição

Array de saída

0	1	2	3	4	5	6	7
0	0		2	3	3	3	

Preencher o *array* de saída, copiando os elementos da entrada de trás para frente nas suas respectivas posições

Array de entrada

0	1	2	3	4	5	6	7
2	5	3	0	2	3	0	3

Array de contagem

0	1	2	3	4	5
0	2	3	4	7	8

Atualizar array de contagem

Array de saída

0	1	2	3	4	5	6	7
0	0		2	3	3	3	

Preencher o *array* de saída, copiando os elementos da entrada de trás para frente nas suas respectivas posições

Array de entrada

0	1	2	3	4	5	6	7
2	5	3	0	2	3	0	3

Array de contagem

0	1	2	3	4	5
0	2	3	4	7	8

Array de saída

0	1	2	3	4	5	6	7
0	0		2	3	3	3	

Preencher o *array* de saída, copiando os elementos da entrada de trás para frente nas suas respectivas posições

Array de entrada

0	1	2	3	4	5	6	7
2	5	3	0	2	3	0	3

Array de contagem

0	1	2	3	4	5
0	2	3	4	7	8

Array de saída

0	1	2	3	4	5	6	7
0	0		2	3	3	3	

Preencher o *array* de saída, copiando os elementos da entrada de trás para frente nas suas respectivas posições

Array de entrada

0	1	2	3	4	5	6	7
2	5	3	0	2	3	0	3

Array de contagem

0	1	2	3	4	5
0	2	3	4	7	8

8ª posição

Array de saída

0	1	2	3	4	5	6	7
0	0		2	3	3	3	

Preencher o *array* de saída, copiando os elementos da entrada de trás para frente nas suas respectivas posições

Array de entrada

0	1	2	3	4	5	6	7
2	5	3	0	2	3	0	3

Array de contagem

0	1	2	3	4	5
0	2	3	4	7	8

8ª posição

Array de saída

0	1	2	3	4	5	6	7
0	0		2	3	3	3	5

Preencher o *array* de saída, copiando os elementos da entrada de trás para frente nas suas respectivas posições

Array de entrada

0	1	2	3	4	5	6	7
2	5	3	0	2	3	0	3

Array de contagem

0	1	2	3	4	5
0	2	3	4	7	7

Atualizar array de
contagem



Array de saída

0	1	2	3	4	5	6	7
0	0		2	3	3	3	5

Preencher o *array* de saída, copiando os elementos da entrada de trás para frente nas suas respectivas posições

Array de entrada

0	1	2	3	4	5	6	7
2	5	3	0	2	3	0	3

Array de contagem

0	1	2	3	4	5
0	2	3	4	7	7

Array de saída

0	1	2	3	4	5	6	7
0	0		2	3	3	3	5

Preencher o *array* de saída, copiando os elementos da entrada de trás para frente nas suas respectivas posições

Array de entrada

0	1	2	3	4	5	6	7
2	5	3	0	2	3	0	3

Array de contagem

0	1	2	3	4	5
0	2	3	4	7	7

Array de saída

0	1	2	3	4	5	6	7
0	0		2	3	3	3	5

Preencher o *array* de saída, copiando os elementos da entrada de trás para frente nas suas respectivas posições

Array de entrada

0	1	2	3	4	5	6	7
2	5	3	0	2	3	0	3

Array de contagem

0	1	2	3	4	5
0	2	3	4	7	7

3ª posição



Array de saída

0	1	2	3	4	5	6	7
0	0		2	3	3	3	5

Preencher o *array* de saída, copiando os elementos da entrada de trás para frente nas suas respectivas posições

Array de entrada

0	1	2	3	4	5	6	7
2	5	3	0	2	3	0	3

Array de contagem

0	1	2	3	4	5
0	2	3	4	7	7

3ª posição



Array de saída

0	1	2	3	4	5	6	7
0	0	2	2	3	3	3	5

Preencher o *array* de saída, copiando os elementos da entrada de trás para frente nas suas respectivas posições

Array de entrada

0	1	2	3	4	5	6	7
2	5	3	0	2	3	0	3

Array de contagem

0	1	2	3	4	5
0	2	2	4	7	7

Atualizar array de contagem

Array de saída

0	1	2	3	4	5	6	7
0	0	2	2	3	3	3	5


Exercício Resolvido 1

- O Counting Sort pode ser aplicado adequadamente na ordenação de strings e números reais?

Exercício Resolvido 1

- O Counting Sort pode ser aplicado adequadamente na ordenação de strings e números reais?

Falso. No caso das *strings*, temos um problema combinatório para identificar a posição de cada *string* no *array* de Contagem. No caso dos números reais, temos infinitos valores entre dois números inteiros.

- Funcionamento básico
- **Algoritmo** 
- Análise do número de movimentações e comparações

Algoritmo

```
void CountingSort(int[] array, int n) {  
    //Array para contar o numero de ocorrencias de cada elemento  
    int[] count = new int[GetMaior(array, n) + 1];  
    int[] ordenado = new int[n];  
  
    //Agora, o count[i] contem o numero de elementos iguais a i  
    for (int i = 0; i < n; count[array[i]]++, i++);  
  
    //Agora, o count[i] contem o numero de elementos menores ou iguais a i  
    for (int i = 1; i < count.Length; count[i] += count[i - 1], i++);  
  
    //Ordenando  
    for (int i = n - 1; i >= 0; ordenado[count[array[i]] - 1] = array[i], count[array[i]]--, i--);  
  
    //Copiar para o array original  
    for (int i = 0; i < n; array[i] = ordenado[i], i++);  
}
```

Algoritmo

```

void CountingSort(int[] array, int n) {
    //Array para contar o numero de ocorrencias de cada elemento
    int[] count = new int[GetMaior(array, n) + 1];
    int[] ordenado =

    int GetMaior(int[] array, int n) {
        int maior = array[0];
        for (int i = 1; i < n; i++) {
            if (maior < array[i])
                maior = array[i];
        }
        return maior;
    }

    //Agora, o count[
    for (int i = 0; i <

    //Agora, o count[
    for (int i = 1; i <

    //Ordenando
    for (int i = n-1; i >= 0; ordenado[count[array[i]]-1] = array[i], count[array[i]]--, i--);

    //Copiar para o array original
    for (int i = 0; i < n; array[i] = ordenado[i], i++);

}

```

Algoritmo

Array de entrada

0	1	2	3	4	5	6	7
2	5	3	0	2	3	0	3

Array de contagem

0	1	2	3	4	5
2	2	4	7	7	8

7ª posição

Array de saída

0	1	2	3	4	5	6	7

//Ordenando

```
for (int i = n-1; i >= 0; ordenado[count[array[i]]-1] = array[i], count[array[i]]--, i--);  
}
```

Algoritmo

Array de entrada

0	1	2	3	4	5	6	7
2	5	3	0	2	3	0	3

Array de contagem

0	1	2	3	4	5
2	2	4	7	7	8

7ª posição

Array de saída

0	1	2	3	4	5	6	7

//Ordenando

```
for (int i = n-1; i >= 0; ordenado[count[array[i]]-1] = array[i], count[array[i]]--, i--);
```

}

7

Algoritmo

Array de entrada

0	1	2	3	4	5	6	7
2	5	3	0	2	3	0	3

Array de contagem

0	1	2	3	4	5
2	2	4	7	7	8

7ª posição

Array de saída

0	1	2	3	4	5	6	7

//Ordenando

```
for (int i = n-1; i >= 0; ordenado[count[array[7]]-1] = array[i], count[array[i]]--, i--);
```

```
}
```

3

Algoritmo

Array de entrada

0	1	2	3	4	5	6	7
2	5	3	0	2	3	0	3

Array de contagem

0	1	2	3	4	5
2	2	4	7	7	8

7ª posição

Array de saída

0	1	2	3	4	5	6	7

//Ordenando

```
for (int i = n-1; i >= 0; ordenado[count[3]-1] = array[i], count[array[i]]--, i--);
```

}

7

Algoritmo

Array de entrada

0	1	2	3	4	5	6	7
2	5	3	0	2	3	0	3

Array de contagem

0	1	2	3	4	5
2	2	4	7	7	8

7ª posição

Array de saída

0	1	2	3	4	5	6	7

//Ordenando

```
for (int i = n-1; i >= 0; ordenado[7 - 1] = array[7], count[array[i]]--, i--);
```

}

6

3

Algoritmo

Array de entrada

0	1	2	3	4	5	6	7
2	5	3	0	2	3	0	3

Array de contagem

0	1	2	3	4	5
2	2	4	7	7	8

7ª posição

Array de saída

0	1	2	3	4	5	6	7
						3	

//Ordenando

```
for (int i = n-1; i >= 0; ordenado[7 - 1] = array[7], count[array[i]]--, i--);
```

}

6

3

Algoritmo

Array de entrada

0	1	2	3	4	5	6	7
2	5	3	0	2	3	0	3

Array de contagem

0	1	2	3	4	5
2	2	4	6	7	8

Array de saída

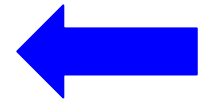
0	1	2	3	4	5	6	7
						3	

//Ordenando

```
for (int i = n-1; i >= 0; ordenado[7 - 1] = array[7], count[array[i]]--, i--);
```

```
}
```

- Funcionamento básico
- Algoritmo
- **Análise do número de movimentações e comparações**



Análise das Operações com Elementos do *Array*

- Para cada elemento do *array* de entrada, incrementá-lo no de contagem
- Fazer com que o *array* de contagem seja acumulativo de tal forma que cada posição i armazene o número de elementos menores ou iguais a i
- Sabendo o número de elementos menores ou iguais a i , preencher o *array* de saída

Análise das Operações com Elementos do *Array*

- Para cada elemento do *array* de entrada, incrementá-lo no de contagem
 $\Theta(n)$
- Fazer com que o *array* de contagem seja acumulativo de tal forma que cada posição i armazene o número de elementos menores ou iguais a i
- Sabendo o número de elementos menores ou iguais a i , preencher o *array* de saída

Análise das Operações com Elementos do *Array*

- Para cada elemento do *array* de entrada, incrementá-lo no de contagem
 $O(n)$
- Fazer com que o *array* de contagem seja acumulativo de tal forma que cada posição i armazene o número de elementos menores ou iguais a i
 $O(n)$
- Sabendo o número de elementos menores ou iguais a i , preencher o *array* de saída

Análise das Operações com Elementos do *Array*

- Para cada elemento do *array* de entrada, incrementá-lo no de contagem

$O(n)$

- Fazer com que o *array* de contagem seja acumulativo de tal forma que cada posição i armazene o número de elementos menores ou iguais a i

$O(n)$

- Sabendo o número de elementos menores ou iguais a i , preencher o *array* de saída

$O(n)$

Análise das Operações com Elementos do *Array*

- Análise da complexidade para operações com elementos do array:

$$\Theta(n) + \Theta(n) + \Theta(n) = \Theta(n)$$